UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/738,542	12/17/2003	Omar Dewan	IDF 2565 (4000-15800)	8268

28003 7590 07/26/2006

SPRINT
6391 SPRINT PARKWAY
KSOPHT0101-Z2100
OVERLAND PARK, KS 66251-2100

EXAMINER	
LU, KUEN S	

ART UNIT	PAPER NUMBER
2167	

DATE MAILED: 07/26/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	10/738,542	DEWAN ET AL.	
	Examiner	Art Unit	
	Kuen S. Lu	2167	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *17 December 2003*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-32* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-32* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *17 December 2003* is/are: a)☐ accepted or b)☒ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date *4/30/04 & 5/6/04*.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. The Action is responsive to Applicant's Application filed December 17, 2003. Claims 1-12 are pending.

Information Disclosure Statement

3. Information Disclosure Statements filed 4/30/2004 and 5/6/2004 are considered and corresponding PTO-1449 forms are electronically signed and attached.

Drawings

4. The informal drawings, filed December 17, 2003 are not of sufficient quality to permit examination. Accordingly, replacement drawing sheets in compliance with 37 CFR 1.121(d) are required in reply to this Office action. The replacement sheet(s) should be labeled "Replacement Sheet" in the page header (as per 37 CFR 1.84(c)) so as not to obstruct any portion of the drawing figures. If the changes are not accepted by the examiner, the applicant will be notified and informed of any required corrective action in the next Office action.

Applicant is given a TWO MONTH time period to submit new drawings in compliance with 37 CFR 1.81. Extensions of time may be obtained under the provisions of 37 CFR 1.136(a). Failure to timely submit replacement drawing sheets will result in ABANDONMENT of the application.

Specification

5. The disclosure is objected to because of the following informalities:

The use of the trademarks JAVA have been noted in this application, for example, Pages 3-4, 13 and 14. It should be capitalized wherever it appears and be

accompanied by the generic terminology.

Although the use of trademarks is permissible in patent applications, the proprietary nature of the marks should be respected and every effort made to prevent their use in any manner which might adversely affect their validity as trademarks.

Appropriate correction is required.

Claim Objections

6. Claim 5 is objected to because of the following informalities:

The word "goup" in the phrase "a goup of service objects" seems to be a typographical error of "group". Appropriate correction is required.

Claim Rejections - 35 USC § 112

7. The following is a quotation of the second paragraph of 35 U.S.C. 112:

The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

7.1. Claims 4, 21-22 and 31-32 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

As per claims 4, 21-22 and 31-32, it contain the trademark/trade names JAVA and Common Object Request Broker Architecture. Where a trademark or trade name is used in a claim as a limitation to identify or describe a particular material or product, the claim does not comply with the requirements of 35 U.S.C. 112, second paragraph. See *Ex parte Simpson*, 218 USPQ 1020 (Bd. App. 1982). The claim scope is uncertain since the trademark or trade name cannot be used properly to identify any particular

material or product. A trademark or trade name is used to identify a source of goods, and not the goods themselves. Thus, a trademark or trade name does not identify or describe the goods associated with the trademark or trade name. In the present case, the trademark/trade name is used to identify/describe an object or an architecture and, accordingly, the identification/description is indefinite.

Claim Rejections - 35 USC § 101

8. 35 U.S.C. § 101 reads as follows:

Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

8.1. Claims 1-32 are rejected under 35 U.S.C. § 101 because the claimed invention is directed to non-statutory subject matter.

As set forth in MPEP 2106 (II) (A):

The claimed invention as a whole must accomplish a practical application. That is, it must produce a "useful, concrete and tangible result." State Street, 149 F.3d at 1373, 47 USPQ2d at 1601-02. The purpose of this requirement is to limit patent protection to inventions that possess a certain level of "real world" value, as opposed to subject matter that represents nothing more than an idea or concept, or is simply a starting point for future investigation or research (Brenner v. Manson, 383 U.S. 519, 528-36, 148 USPQ 689, 693-96); In re Ziegler, 992, F.2d 1197, 1200-03, 26 USPQ2d 1600, 1603-06 (Fed. Cir. 1993)). Accordingly, a complete disclosure should contain some indication of the practical application for the claimed invention, i.e., why the applicant believes the claimed invention is useful.

Apart from the utility requirement of 35 U.S.C. 101, usefulness under the patent eligibility standard requires significant functionality to be present to satisfy the useful result aspect of the practical application requirement. See Arrhythmia, 958 F.2d at 1057, 22 USPQ2d at 1036. Merely claiming nonfunctional descriptive material stored in a computer-readable medium does not make the invention eligible for patenting. For example, a claim directed to a word processing file stored on a disk may satisfy the utility requirement of 35 U.S.C. 101 since the information stored may have some "real world" value. However, the mere fact that the claim may satisfy the utility requirement of 35 U.S.C. 101 does not mean that a useful result is achieved under the practical application requirement. The claimed invention as a whole must produce a "useful, concrete and tangible" result to have a practical application.

As per claims 1-32, the claimed invention represents an abstract service or an abstract methodology of operable modules. The service and method do not produce tangible or useful, for example, there is no useful or tangible result asserted or

established by simply performing steps of maintaining and retuning location of an interface eve if result is returned to application itself. However, tangible, concrete and useful result is required in a practical application test. The consequence is non-statutory.

Claim Rejections - 35 USC § 102

9. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

(e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

9.1. Claims 1-32 are rejected under 35 U.S.C. 102(b) as anticipated by Christof Dallermassl: Aspects of Integration of Heterogeneous Server Systems in Intranets – the Java Approach, Graz University of Technology, Graz, November 1999 (hereafter "Dallermassl").

As per claim 1, Dallermassl teaches "A naming service for locating a service in an enterprise" (See Fig. 4.3 and Page 43 where Voyager ORB offers API allowing objects to communicate with CORBA naming service), comprising:

"a first module operable to maintain a location of an interface, the interface having a reference to a service" (See Fig. 4.3 and Page 43 where Voyager ORB offers API allowing objects to communicate with CORBA naming service); and

"a second module operable to provide the location of the interface to an application in response to receiving a request from the application for the location of the service" (See Fig. 3.3 and Pages 26-27, Para. 3.4.2 where JNDI defines and supports hierarchical structures of objects by using naming and directory services and having objects stored in directory, and a Dino, Distributed Interactive Network Objects, is implemented as an external embedded system being enabled to connect all directory services).

Fig. 4.3 and Page 43 where Voyager ORB offers API allowing objects to communicate with CORBA naming service Fig. 3.3 and Pages 26-27, Para. 3.4.2 where JNDI defines and supports hierarchical structures of objects by using naming and directory services and having objects stored in directory, and a Dino, Distributed Interactive Network Objects, is implemented as an external embedded system being enabled to connect all directory services

As per claim 20, Dallermassl teaches "An enterprise naming service for applications to locate services" (See Fig. 4.3 and Page 43 where Voyager ORB offers API allowing objects to communicate with CORBA naming service), comprising:

"a binding module to associate a first service with a location of an interface maintaining a reference to the first service, the binding module further operable to associate a second

service with a location of the second service" (See Fig. 4.3 and Page 43 where Voyager ORB offers API allowing objects to communicate with CORBA naming service); and

"a look-up module operative to provide the location of the interface in response to a request by an application for the first service, the look-up module further operable to provide the location of the second service in response to a request by a second application" (See Fig. 3.3 and Pages 26-27, Para. 3.4.2 where JNDI defines and supports hierarchical structures of objects by using naming and directory services and having objects stored in directory, and a Dino, Distributed Interactive Network Objects, is implemented as an external embedded system being enabled to connect all directory services).

As per claim 27, Dallermassl teaches "A method for locating a service in an enterprise" (See Fig. 4.3 and Page 43 where Voyager ORB offers API allowing objects to communicate with CORBA naming service), comprising:

"associating a service with a location with an interface maintaining a reference to a service" (See Fig. 4.3 and Page 43 where Voyager ORB offers API allowing objects to communicate with CORBA naming service);

"requesting, by an application desiring to employ the service, the location of the service" (See Fig. 3.3 and Pages 26-27, Para. 3.4.2 where JNDI defines and supports hierarchical structures of objects by using naming and directory services and having objects stored in directory, and a Dino, Distributed Interactive Network Objects, is implemented as an external embedded system being enabled to connect all directory services); and

"returning the location of the interface to the application" (See Fig. 3.3 and Pages 26-27, Para. 3.4.2 where a Dino, Distributed Interactive Network Objects, is implemented as an external embedded system to JNDI and enabled to connect all directory services via JNDI).

As per claim 2, Dallermassl further teaches "wherein application is operable, using the location of the interface, to the service using the interface" (See Fig. 3.3 and Pages 26-27, Para. 3.4.2 where JNDI is the interface, and a Dino is implemented as an external embedded system being enabled applications to connect all directory services).

As per claim 3, Dallermassl further teaches "the service is further defined as a service object" (See Fig. 3.1 and Page 18, last Paragraph where application requests an operation performed by distributed object and result returned in a CORBA architecture).

As per claim 4, Dallermassl further teaches "the service object is further defined as a Java service object and wherein the interface is further defined as a java directory and naming interface" (See Fig. 33, and Pages 20 and 26 where Java objects at remote hosts are invoked by Java application and JDNI is an naming interface).

As per claim 5, Dallermassl further teaches "the service object is selected from a group of service objects including an enterprise Java Bean, a queue, and a queue manger" (See Page 45, Para. 4.3.3. and Page 54, Para. 5.2.1 where application offers Enterprise Java Bean environment and Dino system was message based having events queued in

global message queue and listened by interested components).

As per claim 6, Dallermassl further teaches "the first module id further operable to maintain a second location associates with a second service, the second module further operable to provide the second location to a second application in response to receiving a request for the second service from the second application, the second application using the second location to use the service" (See Fig. 4.3, and Pages 20 and 43 where Voyager ORB offers API to access RMI registry naming service and RMI registry, and RMI enables Java application to invoke method of Java objects on remote hosts).

As per claim 7, Dallermassl further teaches "the second service is a service object and wherein the second location is useful by the application to directly invoke the services" (See Fig. 4.3, and Pages 20 and 43 where RMI enables Java application to invoke method of Java objects on remote hosts).

As per claim 8, Dallermassl further teaches "second location is selected from a group of locations including an address and reference location" (See Fig. 4.3, and Pages 20 and 43 where RMI enables Java applications to invoke methods of Java objects on remote hosts).

As per claim 9, Dallermassl further teaches "first module further maintains an identifier corresponding to the service and associates the identifier with the location of the interface" (See Page 75, Para. 5.4.4 where applications access the Dino system by use of its API and

its views, and internal IDs are used to address objects in the Dino space).

As per claim 10, Dallermassl further teaches "the identifier is a service type of the service" (See Page 27, first paragraph where objects stored in directory are of type).

As per claim 11, Dallermassl further teaches "the identifier includes is a name and a service type associated with the service" (See Pages 27, first paragraph and 75, Para. 5.4.4 where applications access the Dino system by use of its API and its views, and internal IDs are used to address objects in the Dino space and system provides names for the objects).

As per claim 12, Dallermassl further teaches "the interface is a server and the application is a client of the server, the client using the server to provide the service to the client" (See Figs. 3.1-3.2 and Pages 17-18, Para. 3.1.1 and Pages 20-21, Para. 3.1.2 where applications are running at clients and server provides services via interface in the CORBA architecture).

As per claim 13, Dallermassl further teaches "the second module returns meta-information to the application, the meta-information including a reference useful by the application for employing the service" (See Page 20, Para. 3.1.2 where Java application invokes method of remote Java objects and receives reference which is a metadata).

As per claim 14, Dallermassl further teaches "the service is a server maintaining a plurality of classes and a plurality of objects, at least one of the objects useful by the application" (See Page 20, Para. 3.1.2 where Java application invokes method of remote Java objects and

Page 3.3, Para. 3.3 where JDBC is a Java API consists of a set of classes).

As per claim 15, Dallermassl further teaches "first module stores the location of the interface in a datastore" (See Page 3.3, Para. 3.3 where JDBC is a Java API consists of a set of classes and interfaces for accessing a database management system).

As per claim 16, Dallermassl further teaches "the datastore is further defined as a lightweight directory access protocol based datastore" (See Pages 26-27, Para. 3.4.2 where Dino system uses JNDI to connect directory services, including LDAP).

As per claim 17, Dallermassl further teaches "including a third module operable to store a service status information related to the service, the third module operable to search and return the service status information related to the service in response to a request" (See Fig. 4.3, and Pages 20 and 43 where Voyager ORB offers API to access COS naming service).

As per claim 18, Dallermassl further teaches "a hypertext markup language interface is employed to communicate with the third module" (See Page 20, Para. 3.1.2 where RMI use HTTP for network communications).

As per claim 19, Dallermassl further teaches "the third module is defined as a name service browser" (See Fig. 4.3, and Pages 20 and 43 where Voyager ORB offers API to access COS naming service).

As per claim 21, Dallemassl further teaches "the first service is a service object and the second service is a Common Object Request Broker Architecture object" (See Fig. 4.3 and Page 43 where Voyager ORB offers API allowing objects to communicate with CORBA naming service and at Fig. 4.3, and Pages 20 and 43 where Voyager ORB offers API to access RMI registry naming service and RMI registry, and RMI enables Java application to invoke method of Java objects on remote hosts).

As per claim 22, Dallemassl further teaches "the interface is a Java Naming and Directory Interface and wherein the first service is a Java service object" (See Fig. 33, and Pages 20 and 26 where Java objects at remote hosts are invoked by Java application and JDNI is an naming interface).

As per claim 23, Dallemassl further teaches "a name service browser module operable to maintain a service status information related to one of the first and second services, the name service browser operable to search and return the service status information of one of the first and second services in response to a request" (See Fig. 4.3, and Pages 20 and 43 where Voyager ORB offers API to access COS naming service and at Fig. 4.2 and Pages 39-40, Para. 4.2.2 where in CORBA architecture Web browser browsing services responding client request).

As per claim 24, Dallemassl further teaches "the binding module is further operable to maintain a version identifier associated with at least one of the first and second

are maintained" (See Page 14, Para. 2.9 where objects stored in Dino system are version controlled).

As per claim 25, Dallermassl further teaches "the look-up module is further operable to return the location associated to a first version of the first service" (See Page 14, Para. 2.9 where objects stored in Dino system are version controlled, similar to a CVS, and supporting checking in and out, locking, retrieving, tagging and merging of specific version).

As per claim 26, Dallermassl further teaches "the look-up module is further operable to return the location associated to a first version of the second service" (See Page 14, Para. 2.9 where objects stored in Dino system are version controlled and supporting checking in and out, locking, retrieving, tagging and merging of specific version).

As per claim 28, Dallermassl further teaches the following:
"using the location to communication between the application and the interface" (See Fig. 4.3 and Page 43 where Voyager ORB offers API allowing objects to communicate with CORBA naming service);
"requesting, by the application, the service from the interface" (See Fig. 3.3 and Pages 26-27, Para. 3.4.2 where JNDI defines and supports hierarchical structures of objects by using naming and directory services and having objects stored in directory, and a Dino, Distributed Interactive Network Objects, is implemented as an external embedded system being enabled to connect all directory services); and

"using the service by the application" (See Fig. 3.3 and Pages 26-27, Para. 3.4.2 where a Dino, Distributed Interactive Network Objects, is implemented as an external embedded system to JNDI and enabled to connect all directory services via JNDI).

As per claim 29, Dallermassl further teaches "the application uses a service identifier to request the location of the service" (See Page 75, Para. 5.4.4 where applications access the Dino system by use of its API and its views, and internal IDs are used to address objects in the Dino space).

As per claim 30, Dallermassl further teaches "the service is defined as a service object and wherein the interface is further defined as a naming and directory interface" (See Fig. 33, and Pages 20 and 26 where Java objects at remote hosts are invoked by Java application and JDNI is an naming interface).

As per claim 31, Dallermassl further teaches "the interface is defined as a Java Naming Directory Interface and the service is an Enterprise Java Bean, the method further comprising associating an identifier, a version and a second location with a Common Object Request Broker Architecture object" (See Page 45, Para. 4.3.3. and Page 54, Para. 5.2.1 where application offers Enterprise Java Bean environment and Dino system was message based having events queued in global message queue and listened by interested components and at Fig. 4.3 and Page 43 where Voyager ORB offers API allowing objects to communicate with CORBA naming service).

As per claim 32, Dallermassl further teaches the following:

"requesting the second location of the Common Object Request Broker Architecture object using the identifier and version of the Common Object Request Broker Architecture object" (See Fig. 4.3 and Page 43 where Voyager ORB offers API allowing objects to communicate with CORBA naming service and at Page 14, Para. 2.9 where objects stored in Dino system are version controlled);

"returning the second location of the Common Object Request Broker Architecture object based on the identifier and version of the Common Object Request Broker Architecture object" (See Fig. 4.3 and Page 43 where Voyager ORB offers API allowing objects to communicate with CORBA naming service, Page 75, Para. 5.4.4 where applications access the Dino system by use of its API and its views, and internal IDs are used to address objects in the Dino space, and Fig. 3.3 and Pages 26-27, Para. 3.4.2 where JNDI defines and supports hierarchical structures of objects by using naming and directory services and having objects stored in directory, and a Dino, Distributed Interactive Network Objects, is implemented as an external embedded system being enabled to connect all directory services);

"connecting to the Common Object Request Broker Architecture object using the second location" (See Fig. 4.3 and Page 43 where Voyager ORB offers API allowing objects to communicate with CORBA naming service and at Fig. 3.3 and Pages 26-27, Para. 3.4.2 where JNDI defines and supports hierarchical structures of objects by using naming and directory services and having objects stored in directory, and a Dino, Distributed Interactive

Network Objects, is implemented as an external embedded system being enabled to connect all directory services); and

"employing the Common Object Request Broker Architecture object at the second location" (See Fig. 4.3 and Page 43 where Voyager ORB offers API allowing objects to communicate with CORBA naming service and at Fig. 3.3 and Pages 26-27, Para. 3.4.2 where JNDI defines and supports hierarchical structures of objects by using naming and directory services and having objects stored in directory, and a Dino, Distributed Interactive Network Objects, is implemented as an external embedded system being enabled to connect all directory services).

Conclusion

10. The prior art made of record

U. Christof Dallermassl: Aspects of Integration of Heterogeneous Server Systems in Intranets – the Java Approach, Graz University of Technology, Graz, November 1999.

The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

V. Barlen et al.: Implementation and Practical Use of LDAP on the IBM @server iSeries Server, April 2002, IBM

A. U.S. Patent Application 2003/0018701

B. U.S. Patent No. 7,036,127

C. U.S. Patent Application 2005/0015401

D. U.S. Patent No. 5,987,471

Contact Information

11. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Kuen S Lu whose telephone number is (571) 272-4114. The examiner can normally be reached on Monday-Friday (8:00 am-5:00 pm). If attempts to reach the examiner by telephone pre unsuccessful, the examiner's Supervisor, John Cottingham can be reached on (571) 272-7079. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for Page 13 published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 886-217-9197 (toll-free).

Kuen S. Lu
K.S. Lu
Patent Examiner, Art Unit 2167

July 20, 2006